

PROCESSOR WITH UNAUTHORIZED EXECUTION PREVENTING FUNCTION OF PROGRAM, INSTRUCTION TO BE EXECUTED BY PROCESSOR AND UNAUTHORIZED EXECUTION PREVENTING METHOD OF THE PROGRAM

Publication number: JP11345117 (A)

Publication date: 1999-12-14

Inventor(s): OKADA TAKAYUKI +

Applicant(s): IBM +

Classification:

- **international:** **G06F21/00; G06F21/22; G06F21/00; G06F21/22;** (IPC1-7): G06F9/06

- **European:** G06F21/00N7P5H

Application number: JP19980136724 19980519

Priority number(s): JP19980136724 19980519

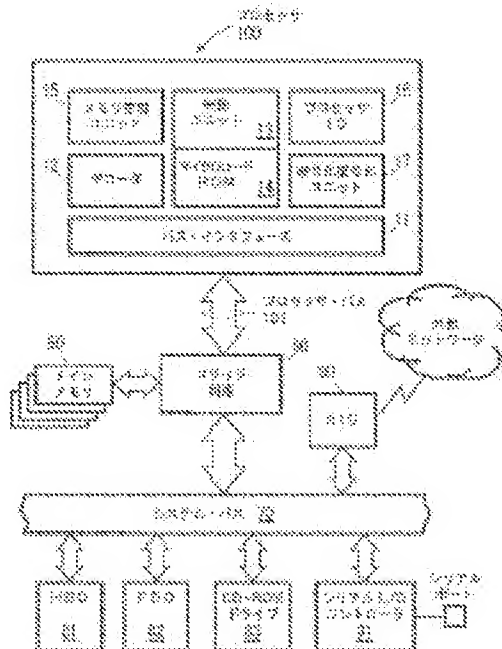
Also published as:

JP3713141 (B2)

US6704872 (B1)

Abstract of JP 11345117 (A)

PROBLEM TO BE SOLVED: To provide an improved techniques for preventing unauthorized use/ execution of a software program provided to a computer system. **SOLUTION:** A processor 100 incorporates a memory managing unit 15 for managing the storage operation of code/data on a main memory and is provided with an execution permitting instruction for permitting the execution of an application on the main memory as an instruction set. The procedure for the execution permitting instruction is defined by a microprogram and provided with a procedure for certifying the right of using a program. Since the certifying operation is executed inside the processor, it is nearly impossible to monitor/ detect operation from the outside.; Besides by using a processor ID 16 characteristically provided in each processor chip as identification information to be used for certifying processing, the right of using the program can be limited to a single processor.



Data supplied from the **espacenet** database — Worldwide

(19) 日本国特許庁 (J P)

(12) 公開特許公報 (A)

(11) 特許出願公開番号

特開平11-345117

(43) 公開日 平成11年(1999)12月14日

(51) Int.Cl.⁶

G 0 6 F 9/06

識別記号

5 5 0

F I

G 0 6 F 9/06

5 5 0 H

審査請求 未請求 請求項の数18 OL (全 21 頁)

(21) 出願番号 特願平10-136724

(22) 出願日 平成10年(1998) 5 月19日

(71) 出願人 390009531

インターナショナル・ビジネス・マシーンズ・コーポレーション

INTERNATIONAL BUSINESS MACHINES CORPORATION

アメリカ合衆国10504、ニューヨーク州

アーモンク (番地なし)

(72) 発明者 岡田 高幸

滋賀県野洲郡野洲町大字市三宅300番地

日本アイ・ビー・エム株式会社 野洲事業所内

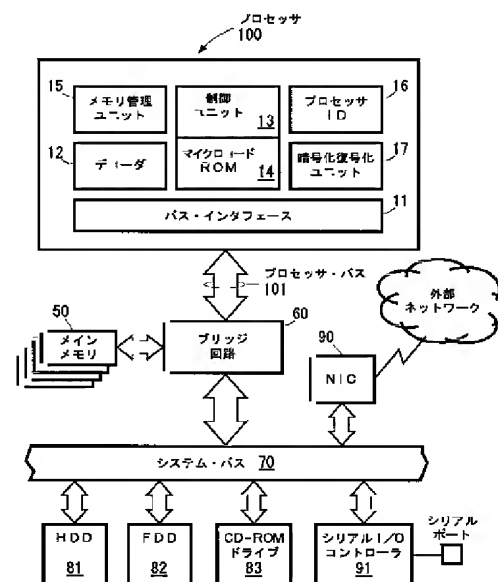
(74) 代理人 弁理士 坂口 博 (外1名)

(54) 【発明の名称】 プログラムの不正実行防止機能付きプロセッサ、プロセッサが実行するインストラクション、及びプログラムの不正実行防止方法

(57) 【要約】 (修正有)

【課題】 コンピュータ・システムに提供されるソフトウェア・プログラムの不正な使用・実行を防止する優れた技術を提供する。

【解決手段】 プロセッサは、メイン・メモリ上のコード／データの格納動作を管理するためのメモリ管理ユニットを内蔵し、かつインストラクション・セットとして、メイン・メモリ上のアプリケーションの実行を許可する実行許可インストラクションを含んでいる。実行許可インストラクションの手続きはマイクロプログラムによって定義され、プログラムの使用权を認証するための手続きを含んでいる。認証動作は、プロセッサ内部で実行されるので、外部から監視・検出することは殆ど不可能である。また、認証処理に用いる識別情報として、プロセッサ・チップが固有に持つプロセッサIDを使用することによって、プログラムの使用权を単一のプロセッサに限定することができる。



【特許請求の範囲】

【請求項1】メイン・メモリを管理する機能を持つメモリ管理ユニットを内蔵するとともに、自身を動作させるためのインストラクション・セットを有するタイプの、プログラムの不正実行防止機能付きプロセッサであって、

前記インストラクション・セット中には前記メイン・メモリ上のプログラム・コードの実行を許可する実行許可インストラクションが含まれ、

前記実行許可インストラクションは、実行すべきプログラムの識別番号とアドレス情報・属性情報を前記メモリ管理ユニットに与える動作の他に、所定の認証動作を含む、ことを特徴とするプログラムの不正実行防止機能付きプロセッサ。

【請求項2】前記所定の認証動作は、前記プロセッサ内部に格納された識別情報と前記プロセッサの外部から供給された識別情報との比較動作を含むことを特徴とする請求項1に記載のプログラムの不正実行防止機能付きプロセッサ。

【請求項3】前記識別情報は、前記プロセッサに固有のプロセッサIDとソフトウェア・プログラムに固有のソフトウェアIDとで構成されることを特徴とする請求項2に記載のプログラムの不正実行防止機能付きプロセッサ。

【請求項4】前記識別情報は、前記プロセッサが内部的に発生した乱数とソフトウェア・プログラムに固有のソフトウェアIDとで構成されることを特徴とする請求項2に記載のプログラムの不正実行防止機能付きプロセッサ。

【請求項5】前記所定の認証動作が成功裡に終わったときには実行すべきプログラムの識別番号とアドレス情報・属性情報を前記メモリ管理ユニットに与える動作を許可するが、前記所定の認証動作が失敗したときには実行すべきプログラムの識別番号とアドレス情報・属性情報を前記メモリ管理ユニットに与える動作を禁止することを特徴とする請求項1に記載のプログラムの不正実行防止機能付きプロセッサ。

【請求項6】メイン・メモリを管理するためのメモリ管理ユニットを内蔵したタイプのプロセッサが実行するインストラクションであって、

前記インストラクションは、アプリケーションの実行を許可するためのインストラクションであり、

所定の認証動作と、

実行すべきプログラムの識別番号とアドレス情報・属性情報を前記メモリ管理ユニットに与える動作と、を特徴とするプロセッサが実行するインストラクション。

【請求項7】前記所定の認証動作は、前記プロセッサ内部に格納された識別情報と前記プロセッサの外部から供給された識別情報との比較動作を含むことを特徴とする請求項6に記載のプロセッサが実行するインストラクシ

ョン。

【請求項8】前記識別情報は、前記プロセッサに固有のプロセッサIDとソフトウェア・プログラムに固有のソフトウェアIDとで構成されることを特徴とする請求項7に記載のプロセッサが実行するインストラクション。

【請求項9】前記識別情報は、前記プロセッサが内部的に発生した乱数とソフトウェア・プログラムに固有のソフトウェアIDとで構成されることを特徴とする請求項7に記載のプロセッサが実行するインストラクション。

【請求項10】前記所定の認証動作が成功裡に終わったときには実行すべきプログラムの識別番号とアドレス情報・属性情報を前記メモリ管理ユニットに与える動作を許可するが、前記所定の認証動作が失敗したときには実行すべきプログラムの識別番号とアドレス情報・属性情報を前記メモリ管理ユニットに与える動作を禁止することを特徴とする請求項6に記載のプロセッサが実行するインストラクション。

【請求項11】メイン・メモリ上にプログラム・コードやデータを読み出し／書き込み動作しながら所定の処理を実行するタイプの、プログラムの不正実行防止機能付きプロセッサであって、

前記プロセッサの外部とインストラクションやデータの授受を行うバス・インターフェースと、

受け取ったインストラクションの解釈を行うデコーダと、

1以上のインストラクションについての手順を格納した内蔵メモリと、

1以上のインストラクションについての手順を実装した論理回路と、

前記内蔵メモリに格納された手順や前記論理回路に実装された手順に従ってインストラクションを実行する制御ユニットと、

前記プロセッサに固有のプロセッサIDと、

前記メイン・メモリを管理する機能を持つメモリ管理ユニットとを含み、

前記内蔵メモリ又は前記論理回路のうちの1つは前記メイン・メモリ上のプログラム・コードの実行を許可する実行許可インストラクションの手順を格納しており、

且つ、前記実行許可インストラクションの手順は、実行すべきプログラムの識別情報とアドレス情報・属性情報を前記メモリ管理ユニットに与える動作の他に、所定の認証動作を含む、ことを特徴とするプログラムの不正実行防止機能付きプロセッサ。

【請求項12】さらに、前記プロセッサIDを暗号化して前記バス・インターフェース経由で前記プロセッサの外部に出力する暗号化手段と、暗号化された識別情報を前記バス・インターフェース経由で前記プロセッサの外部から受け取るとともに復号化してプロセッサIDとソフトウェアIDを取り出す復号化手段とを含み、前記所定の認証動作は各ID同士の比較動作を含むこと

を特徴とする請求項11に記載のプログラムの不正実行防止機能付きプロセッサ。

【請求項13】前記所定の認証動作が成功裡に終わったときには実行すべきプログラムの識別番号とアドレス情報・属性情報を前記メモリ管理ユニットに与える動作を許可するが、前記所定の認証動作が失敗したときには実行すべきプログラムの識別番号とアドレス情報・属性情報を前記メモリ管理ユニットに与える動作を禁止することを特徴とする請求項11に記載のプログラムの不正実行防止機能付きプロセッサ。

【請求項14】メイン・メモリ上にプログラム・コードやデータを読み出し／書き込み動作しながら所定の処理を実行するタイプの、プログラムの不正実行防止機能付きプロセッサであって、
前記プロセッサの外部とインストラクションやデータの授受を行うバス・インターフェースと、
受け取ったインストラクションの解読を行うデコーダと、

1以上のインストラクションについての手順を格納した内蔵メモリと、

1以上のインストラクションについての手順を実装した論理回路と、

前記内蔵メモリに格納された手順や前記論理回路に実装された手順に従ってインストラクションを実行する制御ユニットと、

乱数発生ユニットと、

前記メイン・メモリを管理する機能を持つメモリ管理ユニットとを含み、

前記内蔵メモリは前記メイン・メモリ上のプログラム・コードの実行を許可する実行許可インストラクションの手順を格納しており、

且つ、前記実行許可インストラクションの手順は、実行すべきプログラムの識別情報とアドレス情報・属性情報を前記メモリ管理ユニットに与える動作の他に、所定の認証動作を含む、ことを特徴とするプログラムの不正実行防止機能付きプロセッサ。

【請求項15】さらに、前記乱数発生ユニットが発生する乱数を前記バス・インターフェース経由で前記プロセッサの外部に出力する乱数出力手段と、暗号化された識別情報を前記バス・インターフェース経由で前記プロセッサの外部から受け取るとともに復号化して乱数とソフトウェアIDを取り出す復号化手段とを含み、
前記所定の認証動作は乱数及びソフトウェアIDの比較動作を含むことを特徴とする請求項14に記載のプログラムの不正実行防止機能付きプロセッサ。

【請求項16】前記所定の認証動作が成功裡に終わったときには実行すべきプログラムの識別番号とアドレス情報・属性情報を前記メモリ管理ユニットに与える動作を許可するが、前記所定の認証動作が失敗したときには実行すべきプログラムの識別番号とアドレス情報・属性情

報を前記メモリ管理ユニットに与える動作を禁止することを特徴とする請求項14に記載のプログラムの不正実行防止機能付きプロセッサ。

【請求項17】ソフトウェア供給者がソフトウェア・プログラムの使用権をただ1つのプロセッサにのみ許可するための、プログラムの不正実行防止方法であって、

(a)ソフトウェアIDが付されたソフトウェア・プログラムを受容するステップと、(b)前記プロセッサに固有のプロセッサIDを暗号化して、前記プロセッサの外部に出力するステップと、(c)ソフトウェア供給者において、暗号化されたプロセッサIDを一旦復号化して、ソフトウェア供給者自身が管理するソフトウェアIDを前記プロセッサIDとともに暗号化するステップと、(d)ステップ(c)によって暗号化した識別情報を前記プロセッサに供給するステップと、(e)前記プロセッサにおいて、識別情報を復号化してプロセッサIDとソフトウェアIDを取り出すステップと、(f)前記プロセッサが保持するプロセッサID及び前記ソフトウェア・プログラムに付されたソフトウェアIDを、ステップ(e)で得られたプロセッサIDとソフトウェアIDの各々と照合するステップと、(g)前記照合処理が成功裡に終わったときのみ、前記プロセッサが前記ソフトウェア・プログラムを実行することを許可するステップと、を具備することを特徴とするプログラムの不正実行防止方法。

【請求項18】ソフトウェア供給者がソフトウェア・プログラムの使用権をただ1つのプロセッサにのみ許可するための、プログラムの不正実行防止方法であって、

(a)ソフトウェアIDが付されたソフトウェア・プログラムを受容するステップと、(b)前記プロセッサが乱数を発生して前記プロセッサの外部に出力するステップと、(c)ソフトウェア供給者において、ソフトウェア供給者自身が管理するソフトウェアIDを前記プロセッサが出力した乱数とともに暗号化するステップと、

(d)ステップ(c)によって暗号化した識別情報を前記プロセッサに供給するステップと、(e)前記プロセッサにおいて、識別情報を復号化して乱数とソフトウェアIDを取り出すステップと、(f)前記プロセッサが保持する乱数及び前記ソフトウェア・プログラムに付されたソフトウェアIDを、ステップ(e)で得られた乱数とソフトウェアIDの各々と照合するステップと、

(g)前記照合処理が成功裡に終わったときのみ、前記プロセッサが前記ソフトウェア・プログラムを実行することを許可するステップと、を具備することを特徴とするプログラムの不正実行防止方法。

【発明の詳細な説明】

【0001】

【発明の属する技術分野】本発明は、コンピュータ・システムに提供されるソフトウェア・プログラムの不正な使用・実行を防止する技術に係り、特に、プログラムの

不正実行防止機能を備えたプロセッサ、プログラムの不正実行を防止するためにプロセッサが実行するインストラクション、及びプログラムの不正実行防止方法に関する。更に詳しくは、本発明は、外部から監視・変更することができないプログラムの不正実行防止機能付きプロセッサ、プロセッサが実行するインストラクション、及びプログラムの不正実行防止方法に関する。

【0002】

【従来の技術】昨今の技術革新に伴い、コンピュータ・システムは、大学・研究機関の他、企業や一般家庭にも広範に浸透してきている。コンピュータ・システムには、ホスト/メインフレームやオフィス・コンピュータ以外に、比較的安価で一般ユーザも購入可能なワークステーション、パーソナル・コンピュータ（PC）などが挙げられる。特に、PCの急速な普及には目覚ましいものがある。

【0003】このような技術動向下では、コンピュータ・システム上で稼働する各種ソフトウェアも開発・市販されており、ソフトウェア産業もハードウェア産業に匹敵しあるいは凌駕する隆盛を遂げている。ここで言うソフトウェアには、コンピュータ・システム全体の動作を制御するための「OS（オペレーティング・システム）」や、システム上で各ユーザの目的・業務に適った機能オペレーションを実現するための「アプリケーション」が含まれる。アプリケーション・プログラムは、ワープロ、スプレッドシート、データベース、通信など各種機能・用途のものが揃っている。

【0004】ユーザは、自身のコンピュータ・システム上に導入したいソフトウェアを、例えばフロッピー・ディスクやCD-ROMなどのリムーバブル・メディアの形態で購入し、これらメディアを適切なドライブ・ユニットに装填して用いるのが一般的である。また、最近では、インターネットなどの外部ネットワークを経由して、所望のソフトウェアを自身のローカル・ディスクにファイル転送（ダウンロード）する、という形態によってもソフトウェア・プログラムを導入することが可能となっている。

【0005】但し、ソフトウェア・プログラムを格納したメディアの購入行為は、ソフトウェア・プログラムの限定的な使用権を得たことに過ぎず、購入者はソフトウェア・プログラムの著作権や複製権までも手に入れた訳ではない。ソフトウェア・プログラムの無制限な若しくは無断な複製は、著作権法上、厳に禁止されている行為である（周知）。（なお、ネットワーク経由でダウンロードしたソフトウェア・プログラムについても、その使用は限定的に解釈するのが妥当であろう。）

【0006】また、ソフトウェア・プログラムを提供するソフトウェア・ベンダーの立場から言えば、有料でプログラム格納メディアを配布することによって収益をあげているのであり、無制限で無断な複製行為が横行して

いては、本来享受すべき利潤を担保することができない。ソフトウェア・プログラムの不正な使用行為は、ソフトウェア産業に従事する者の開発意欲をも減退させるものであり、ひいては当該産業界の活動自体を沈静化させかねない。このため、ソフトウェア・プログラムの不正使用や無断な複製を禁止する（あるいは未然に防止する）というプロテクションに関する技術が、従来より開発されている。

【0007】プロテクションの一例は、不正使用を防止したいソフトウェア・プログラムを暗号化してしまうことである。すなわち、ソフトウェア・ベンダーは、暗号化した状態のプログラム格納メディアを有償若しくは無償で配布するとともに、正当なユーザに対してのみ、暗号を解くための鍵を与えることで、メディアに格納されたプログラムの使用を好適に限定することができる訳である。

【0008】ここで、いかなる媒介によってユーザに鍵を与えるかが問題になる。例えばソフトウェア・ベンダーは、封書などの郵便物によってユーザに鍵を与えることも可能であるが、この場合は鍵の不正使用はいとも簡単である。

【0009】さらに技術的に高度な方式として、いわゆる「セキュリティ・デバイス」を用いることが挙げられる。この場合、ソフトウェア・ベンダーは、セキュリティ・デバイスを添付してプログラム格納メディアを流通・配布する。セキュリティ・デバイスは、所定の認証手続に必要な識別情報などを含んでおり、例えばユーザのPCのシリアル・ポート又はパラレル・ポートに接続して用いられる。他方、メディアに格納されたプログラムの中には、プログラム自体の機能・用途に特化したオペレーションを行うための本ルーチンの他に、幾つかのチェック・ポイント（すなわち「認証ルーチン」）を含んでいる。

【0010】認証ルーチンの一例は、各チェック・ポイントにおいて、プログラムを実行中のCPU（Central Processing Unit）がセキュリティ・デバイスにアクセスして識別情報を読み出し、これがプログラムの持つ識別情報と一致するかどうかを照合して、一致しているときのみCPUがプログラムを続行することを許可する、というものである。

【0011】また、認証ルーチンの他の例は次の通りである。すなわち、プログラム実行中のCPUがある規則に従ったコードをセキュリティ・デバイスに書き込む。セキュリティ・デバイスでは、受け取ったコードを用いてスクランブル又は暗号化した識別情報を用意しておく。さらに、CPUはスクランブル又は暗号化された識別情報を読み出し、これをプログラムによってデスクランブル又は復号化してプログラムが所有する識別情報と照合する。照合が成功裡に終わったときのみ、CPUによるプログラムの続行が許可される。CPUとセキュリ

ティ・デバイスの間は、CPUチップ外部のバスを介して接続されているので、スクランブルや暗号化を行うことによって、バス・スヌープによる認証ルーチンの解読を防止することができる。

【0012】これらセキュリティ・デバイスを用いた2つの例は、プログラム実行中のCPUに対する命令（インストラクション）によって認証処理が行われる、という点では共通している。

【0013】セキュリティ・デバイスを用いることにより、セキュリティ・レベルをかなりの程度向上させることができよう。また、ソフトウェア・プログラムの使用権をセキュリティ・デバイスを装着したただ1つのコンピュータ・システムのみ限定することができよう。しかしながら、この方式とて万全とはいえない。例えば、ロジック・アナライザ（周知）を用いれば、バス・トランザクションを容易にモニタすることができるので、認証ルーチンを解析して、識別情報や暗号の鍵の正体を見破ることもできよう。

【0014】さらにセキュリティ・レベルを向上させるために、セキュリティ・デバイスへの入力値を逐次変更する、という対策も採られている。しかしながら、ICE（InCircuitEmulator）/**などの装置を用いればソフトウェアを比較的容易に逆アセンブルすることができ、そうならばセキュリティ・デバイスへのアクセス・ルーチン（すなわち認証ルーチン）を無条件ジャンプするようにプログラムを改ざんすることも可能である。

【0015】本発明者の考察によれば、ソフトウェア・プログラムのプロテクションを強化するためには、まず第一に、セキュリティ動作がプログラム実行に必須であるようにデザインすることが好ましい。この意味において、メディアに格納されたプログラム自身が認証ルーチンを含んでいるという上記従来例は適合している。さらに、この認証ルーチンは、外部からのアクセス・変更が不可能な形態でなければならない。セキュリティ・デバイスを用いた上記従来例の場合、認証ルーチンはシステム・オペレーションの形態で実装され、外観には現れないが、技術的には判読が可能である（前述）。要言すれば、外部アクセス可能な従来のプロテクション方式では、プログラムの保護は万全とはいえないのである。

【0016】《注釈》ICEは、通常はプログラムやハードウェア開発を支援するために使用される装置であり、CPUの代わりにCPUソケットに差すことによって、CPUと全く同じ動作をするようになっている。ICEは、CPUと違って、バスのアクセスやレジスタの値をインストラクション毎に読み出すことが可能である。このようなICEのインストラクション・トレース機能を用いれば、外部デバイスのアドレスをトリガにしてソフトウェア・プログラムの中から認証ルーチンを探し出すことができる。さらに、プログラム・アドレスに違法なパッチを当てれば、セキュリティ・デバイスにア

クセスせずにプログラム実行を継続するように改ざんすることができる。

【0017】

【発明が解決しようとする課題】本発明の目的は、コンピュータ・システムに提供されるソフトウェア・プログラムの不正な使用・実行を防止する優れた技術を提供することにある。

【0018】本発明の更なる目的は、プログラムの不正実行防止機能を備えた優れたプロセッサ、プログラムの不正実行を防止するためにプロセッサが実行するインストラクション、及びプログラムの不正実行防止方法を提供することにある。

【0019】本発明の更なる目的は、外部から監視・変更することができないプログラムの不正実行防止機能付き優れたプロセッサ、プロセッサが実行するインストラクション、及びプログラムの不正実行防止方法を提供することにある。

【0020】

【課題を解決するための手段】本発明は、上記課題を参照してなされたものであり、その第1の側面は、メイン・メモリを管理する機能を持つメモリ管理ユニットを内蔵するとともに、自身を動作させるためのインストラクション・セットを有するタイプの、プログラムの不正実行防止機能付きプロセッサであって、前記インストラクション・セット中には前記メイン・メモリ上のプログラム・コードの実行を許可する実行許可インストラクションが含まれ、前記実行許可インストラクションは、実行すべきプログラムの識別番号とアドレス情報・属性情報を前記メモリ管理ユニットに与える動作の他に、所定の認証動作を含む、ことを特徴とするプログラムの不正実行防止機能付きプロセッサである。

【0021】ここで、前記所定の認証動作は、前記プロセッサ内部に格納された識別情報と前記プロセッサの外部から供給された識別情報との比較動作を含んでもよい。

【0022】認証に用いられる識別情報は、前記プロセッサに固有のプロセッサIDとソフトウェア・プログラムに固有のソフトウェアIDとで構成されていてもよい。

【0023】あるいは、識別情報は、前記プロセッサが内部的に発生した乱数とソフトウェア・プログラムに固有のソフトウェアIDとで構成されていてもよい。

【0024】また、前記所定の認証動作が成功裡に終わったときには実行すべきプログラムの識別番号とアドレス情報・属性情報を前記メモリ管理ユニットに与える動作を許可するが、前記所定の認証動作が失敗したときには実行すべきプログラムの識別番号とアドレス情報・属性情報を前記メモリ管理ユニットに与える動作を禁止するようにしてもよい。

【0025】また、本発明の第2の側面は、メイン・メ

メモリを管理するためのメモリ管理ユニットを内蔵したタイプのプロセッサが実行するインストラクションであって、前記インストラクションは、アプリケーションの実行を許可するためのインストラクションであり、所定の認証動作と、実行すべきプログラムの識別番号とアドレス情報・属性情報を前記メモリ管理ユニットに与える動作と、を特徴とするプロセッサが実行するインストラクションである。

【0026】ここで、前記所定の認証動作は、前記プロセッサ内部に格納された識別情報と前記プロセッサの外部から供給された識別情報との比較動作を含んでいてもよい。

【0027】認証に用いられる識別情報は、前記プロセッサに固有のプロセッサIDとソフトウェア・プログラムに固有のソフトウェアIDとで構成されていてもよい。

【0028】あるいは、識別情報は、前記プロセッサが内部的に発生した乱数とソフトウェア・プログラムに固有のソフトウェアIDとで構成されていてもよい。

【0029】また、前記所定の認証動作が成功裡に終わったときには実行すべきプログラムの識別番号とアドレス情報・属性情報を前記メモリ管理ユニットに与える動作を許可するが、前記所定の認証動作が失敗したときには実行すべきプログラムの識別番号とアドレス情報・属性情報を前記メモリ管理ユニットに与える動作を禁止するようにしてもよい。

【0030】また、本発明の第3の側面は、メイン・メモリ上にプログラム・コードやデータを読み出し／書き込み動作しながら所定の処理を実行するタイプの、プログラムの不正実行防止機能付きプロセッサであって、前記プロセッサの外部とインストラクションやデータの授受を行うバス・インターフェースと、受け取ったインストラクションの解釈を行うデコーダと、1以上のインストラクションについての手順を格納した内蔵メモリと、1以上のインストラクションについての手順を実装した論理回路と、前記内蔵メモリに格納された手順や前記論理回路に実装された手順に従ってインストラクションを実行する制御ユニットと、前記プロセッサに固有のプロセッサIDと、前記メイン・メモリを管理する機能を持つメモリ管理ユニットとを含み、前記内蔵メモリ又は前記論理回路のうちの1つは前記メイン・メモリ上のプログラム・コードの実行を許可する実行許可インストラクションの手順を格納しており、且つ、前記実行許可インストラクションの手順は、実行すべきプログラムの識別情報とアドレス情報・属性情報を前記メモリ管理ユニットに与える動作の他に、所定の認証動作を含む、ことを特徴とするプログラムの不正実行防止機能付きプロセッサ。

【0031】第3の側面に係る不正実行防止機能付きプロセッサは、さらに、前記プロセッサIDを暗号化して

前記バス・インターフェース経由で外部に出力する暗号化手段と、暗号化された識別情報を前記バス・インターフェース経由で外部から受け取るとともに復号化してプロセッサIDとソフトウェアIDを取り出す復号化手段とを含み、前記所定の認証動作は各ID同士の比較動作を含んでいてもよい。

【0032】また、前記所定の認証動作が成功裡に終わったときには実行すべきプログラムの識別番号とアドレス情報・属性情報を前記メモリ管理ユニットに与える動作を許可するが、前記所定の認証動作が失敗したときには実行すべきプログラムの識別番号とアドレス情報・属性情報を前記メモリ管理ユニットに与える動作を禁止するようにしてもよい。

【0033】また、本発明の第4の側面は、メイン・メモリ上にプログラム・コードやデータを読み出し／書き込み動作しながら所定の処理を実行するタイプの、プログラムの不正実行防止機能付きプロセッサであって、前記プロセッサの外部とインストラクションやデータの授受を行うバス・インターフェースと、受け取ったインストラクションの解釈を行うデコーダと、1以上のインストラクションについての手順を格納した内蔵メモリと、1以上のインストラクションについての手順を実装した論理回路と、前記内蔵メモリに格納された手順や前記論理回路に実装された手順に従ってインストラクションを実行する制御ユニットと、乱数発生ユニットと、前記メイン・メモリを管理する機能を持つメモリ管理ユニットとを含み、前記内蔵メモリ又は前記論理回路のうちの1つは前記メイン・メモリ上のプログラム・コードの実行を許可する実行許可インストラクションの手順を格納しており、且つ、前記実行許可インストラクションの手順は、実行すべきプログラムの識別情報とアドレス情報・属性情報を前記メモリ管理ユニットに与える動作の他に、所定の認証動作を含む、ことを特徴とするプログラムの不正実行防止機能付きプロセッサである。

【0034】第4の側面に係る不正実行防止機能付きプロセッサは、さらに、前記乱数発生ユニットが発生する乱数を前記バス・インターフェース経由で外部に出力する乱数出力手段と、暗号化された識別情報を前記バス・インターフェース経由で外部から受け取るとともに復号化して乱数とソフトウェアIDを取り出す復号化手段とを含み、前記所定の認証動作は乱数及びソフトウェアIDの比較動作を含んでいてもよい。

【0035】また、前記所定の認証動作が成功裡に終わったときには実行すべきプログラムの識別番号とアドレス情報・属性情報を前記メモリ管理ユニットに与える動作を許可するが、前記所定の認証動作が失敗したときには実行すべきプログラムの識別番号とアドレス情報・属性情報を前記メモリ管理ユニットに与える動作を禁止するようにしてもよい。

【0036】また、本発明の第5の側面は、ソフトウェ

ア供給者がソフトウェア・プログラムの使用権をただ1つのプロセッサにのみ許可するための、プログラムの不正実行防止方法であって、(a)ソフトウェアIDが付されたソフトウェア・プログラムを受容するステップと、(b)前記プロセッサに固有のプロセッサIDを暗号化して、前記プロセッサの外部に出力するステップと、(c)ソフトウェア供給者において、暗号化されたプロセッサIDを一旦復号化して、ソフトウェア供給者自身が管理するソフトウェアIDを前記プロセッサIDとともに暗号化するステップと、(d)ステップ(c)によって暗号化した識別情報を前記プロセッサに供給するステップと、(e)前記プロセッサにおいて、識別情報を復号化してプロセッサIDとソフトウェアIDを取り出すステップと、(f)前記プロセッサが保持するプロセッサID及び前記ソフトウェア・プログラムに付されたソフトウェアIDを、ステップ(e)で得られたプロセッサIDとソフトウェアIDの各々と照合するステップと、(g)前記照合処理が成功裡に終わったときのみ、前記プロセッサが前記ソフトウェア・プログラムを実行することを許可するステップと、を具備することを特徴とするプログラムの不正実行防止方法である。

【0037】また、本発明の第6の側面は、ソフトウェア供給者がソフトウェア・プログラムの使用権をただ1つのプロセッサにのみ許可するための、プログラムの不正実行防止方法であって、(a)ソフトウェアIDが付されたソフトウェア・プログラムを受容するステップと、(b)前記プロセッサが乱数を発生して前記プロセッサの外部に出力するステップと、(c)ソフトウェア供給者において、ソフトウェア供給者自身が管理するソフトウェアIDを前記プロセッサが出力した乱数とともに暗号化するステップと、(d)ステップ(c)によって暗号化した識別情報を前記プロセッサに供給するステップと、(e)前記プロセッサにおいて、識別情報を復号化して乱数とソフトウェアIDを取り出すステップと、(f)前記プロセッサが保持する乱数及び前記ソフトウェア・プログラムに付されたソフトウェアIDを、ステップ(e)で得られた乱数とソフトウェアIDの各々と照合するステップと、(g)前記照合処理が成功裡に終わったときのみ、前記プロセッサが前記ソフトウェア・プログラムを実行することを許可するステップと、を具備することを特徴とするプログラムの不正実行防止方法である。

【0038】

【作用】本発明に係るプロセッサは、メイン・メモリ上のコード／データの格納動作を管理するためのメモリ管理ユニットを内蔵するタイプであり、且つ、インストラクション・セットの1つとして、メイン・メモリ上のプログラム・コード（アプリケーション）の実行を許可する実行許可インストラクションを含んでいる。

【0039】この種の実行許可インストラクションは、一般には、実行すべきプログラムの識別番号と、プログラムの格納アドレス、プログラムの属性情報をメモリ管理ユニットにセットする、という手続（例えば“LOAD MMU…”というコードで記述される）を含んでいる。インストラクションの手続はマイクロプログラムによって定義される。

【0040】本発明では、この実行許可インストラクションは、メモリ管理ユニットに所定のデータをセットする手続の他に、プログラムの使用権を認証するための認証手続を含んでいる。この認証動作は、例えば、プロセッサ内部で保持された識別情報と、プロセッサの外部（例えばソフトウェア・プログラムの供給者）から供給された識別情報との比較によって行われる。そして、この認証処理が成功裡に終わったときのみ、ソフトウェア・プログラムの実行が許可される。

【0041】認証手続に用いる識別情報は、例えば、プロセッサに固有（例えば製造時に与えられる）プロセッサIDと、ソフトウェア・プログラムに固有に与えられるソフトウェアIDとで構成される。あるいは、識別情報は、プロセッサが時々刻々発生する乱数と、ソフトウェア・プログラムに固有に与えられるソフトウェアIDとで構成される。

【0042】このような認証動作は、プロセッサ内部で実行されるので、外部から監視・検出することは殆ど不可能である。また、認証手続は、プログラムの実行許可を意味するインストラクションを記述したマイクロプログラムの中に含まれているので、そもそも認証手続の存在自体が外部から隠された格好となる。したがって、本発明によればプログラムの不正実行の防止を大幅に強化することができる。

【0043】また、認証処理に用いる識別情報として、プロセッサ・チップが固有に持つプロセッサIDを使用することによって、ある特定のソフトウェア・プログラムの使用権を単一のプロセッサに限定することができる。この意味においても、プログラムの不正実行は強固に禁止されていると言えよう。

【0044】本発明のさらに他の目的、特徴や利点は、後述する本発明の実施例や添付する図面に基づくより詳細な説明によって明らかになるであろう。

【0045】

【発明の実施の形態】以下、図面を参照しながら本発明の実施例を詳解する。

【0046】第1の実施の形態

図1には、本発明の第1の実施形態に係るコンピュータ・システム200及びプロセッサ100のハードウェア構成を模式的に示している。コンピュータ・システム200は、システム200の中核的な制御を行うプロセッサ100の外に、メイン・メモリ50や各種周辺装置を含んでいる。以下、各部について説明する。

【0047】プロセッサ100の外部ピンに直結したプロセッサ・バス101は、ブリッジ回路60経由でシステム・バス70に接続されている。ブリッジ回路60は、プロセッサ・バス101とシステム・バス70間の動作速度の相違を吸収するためのデータ・バッファや、メイン・メモリ50へのアクセス動作を制御するためのメモリ・コントローラを含んでいる。

【0048】メイン・メモリ50は、プロセッサ100が実行するプログラム・コードや処理データを一時格納するための読み書き可能メモリであり、通常は、1以上のDRAM（ダイナミックRAM）チップで構成される。メイン・メモリ50上で展開されるメモリ空間は、メモリ管理ユニット15（後述）によって管理される。

【0049】システム・バス70には、PCI（Peripheral Component Interconnect）バスやISA（Industry Standard Architecture）バスが該当し、各種周辺装置が相互接続されている。周辺装置には、例えばキーボードやディスプレイなどのコンソール類（図示しない）、ハード・ディスク・ドライブ（HDD）81、フロッピー・ディスク・ドライブ（FDD）82、CD-ROMドライブ83などの外部記憶装置類、外部ネットワークと接続するためのネットワーク・インターフェース・カード（NIC）90、シリアル・ポート経由でシリアル・データの入出力を行うためのシリアルI/Oコントローラ91などが含まれる。なお、バス70上の動作は、ロジック・アナライザなどを用いて解析することができる。

【0050】ソフトウェア供給業者は、通常、フロッピー・ディスクやCD-ROMなどの記憶媒体の形態でソフトウェア・プログラムを一般ユーザに配布する。各ユーザは、これら記憶媒体を所定のドライブ・ユニットに装填し、HDD81にコピーすることによって、ソフトウェア・プログラムをシステム200にインストールすなわち利用可能な状態に置くようになっている。また、最近では、インターネットなどの外部ネットワーク経由でソフトウェア・プログラムが配布されることも多くなっている（周知）。

【0051】プロセッサ100は、当業界においてCPU（Central Processing Unit）若しくはMPU（Micro Processing Unit）と呼ばれるものである。プロセッサ100は、通常はマザー・ボード（図示しない）上に搭載される回路チップであり、各入出力装置（図示しない）を制御してデータの授受を行ったり、データを演算処理し、処理結果をメイン・メモリ50上に記憶したりするなどのコンピュータ・システム200の中核機能を果たしている。

【0052】図1に示すように、本実施形態に係るプロセッサ100は、バス・インターフェース11と、デコーダ12と、制御ユニット13と、マイクロコードROM14と、メモリ管理ユニット15と、プロセッサID

16と、暗号化復号化ユニット17とを含んでいる。

【0053】バス・インターフェース11は、プロセッサ100外部のコンポーネントととの間でインストラクションやデータを授受するためのユニットである。バス・インターフェース11は、プロセッサ・バス17と直結している。なお、プロセッサ・バス17は、ブリッジ回路60を経由してメイン・メモリ50やシステム・バス70と相互接続されている。

【0054】デコーダ12は、バス・インターフェース11経由で受け取ったインストラクションを解釈して、制御ユニット13が理解できる形式に変換するためのユニットである。

【0055】制御ユニット13は、インストラクションを実行する手順を整えるユニットである。実行するインストラクションは、ワイヤード・ロジック（配線論理）で実行するタイプと、マイクロ・プログラムで実行するタイプの2つに大別される。前者の場合は、インストラクションの実行制御に必要な論理を直線的に実装した順序回路が用いられる。また、後者の場合には、インストラクションはマイクロコード（又はピココード）と呼ばれるさらに細かいコードの集合（すなわちマイクロプログラム）で構成されており、該当するマイクロプログラムをマイクロコードROM14から読み出して実行することになる。なお、1つのプロセッサで利用できるインストラクション全体を「インストラクション・セット」と呼ぶ。

【0056】マイクロコードROM14は、インストラクションの実行に必要なマイクロコードを格納した読み出し専用メモリである。言い換えれば、マイクロコードROM14は、各インストラクションの実行手順を定義した不揮発メモリである。例えば、“LOAD”、“STORE”、“MOVE”などのインストラクションは、ある変数をプロセッサ100内のレジスタ（図示しない）に書き込むためのインストラクションであるが、これらの動作の詳細をマイクロコードROM14内で記述することによって定義することができる。

【0057】メモリ管理ユニット（MMU）15は、プログラム・コードやデータをメイン・メモリ50上に展開するなど、プロセッサ100のメモリ空間を管理するためのユニットである。メモリ管理ユニット15は、通常、仮想記憶機能とメモリ保護機能という2つの機能を提供している。仮想記憶機能とは、論理アドレスを物理アドレスに変換したり、物理アドレスに該当するブロック（ページ）がメイン・メモリ上に存在しないときに外部記憶装置との間でブロック（ページ）の入れ替え（すなわち「スワッピング」）を行う機能のことである。また、メモリ保護機能とは、1つのプログラムで生じた障害が他のプログラムに影響を及ぼさないようにする機能のことである。

【0058】また、メモリ管理ユニット15は、メモリ

空間上に存在する各プログラムのアドレス情報や属性情報を管理するためのMMUテーブルを備えている。図2には、MMUテーブルの構造や機能を模式的に図解している。MMUテーブルの各エントリは、該当するプログラムのアドレスとフラグを格納する各フィールドを含んでいる。フラグはプログラムの属性情報（例えばプログラムが実行形式か、読み出し専用か、読み書き可能か）を示している。

【0059】メモリ空間上の各プログラムは、夫々に固有のプログラムIDを有している。あるプログラムをプ

Mov R1,Program_ID

（レジスタR1にプログラムIDをセットせよ）

Mov R2,Memory_Address_&Flag

（レジスタR2にアドレスとフラグをセットせよ）

Load MMU, R1,R2

（メモリ管理ユニットにレジスタR1とR2の内容をセットせよ）

Jump Application

（アプリケーションにジャンプせよ）

【0061】上記コード中の“Mov”や“Load”などは、プロセッサ100に対するインストラクションであり、レジスタへの変数のセット等を意味する。各インストラクションの詳細な実行手順は、ワイヤード・ロジック又はマイクロコードROM14で定義されたマイクロプログラムのいずれかの形態で定義されている（上述）。

【0062】上記疑似コードのうち“Load MMU …”なるステップは、実質上、アプリケーションの実行を許可するためのインストラクションである。本実施例では、この“Load MMU …”インストラクションは、マイクロプログラムという形態で実装され、メモリ管理ユニット15への変数セットの他に、所定の認証手続が含まれている。

【0063】この認証手続は、マイクロプログラム中で記述されているので、プロセッサ100の外部からは認証手続の存在自体が見えない。認証手続は、プロセッサ・チップ100内部で完結的に実行されるので、この動作を外部から監視することは殆ど不可能である。したがって、“Load MMU …”インストラクションを発行した外部プログラムからは全く見えない状態で、プロセッサ100はこの認証手続を実行することができる。但し、認証手続の詳細は後述する。

【0064】プロセッサID16は、プロセッサ100に固有の識別情報（通常はシリアルな番号で構成される）であり、例えば製造時に一義的に付与される。プロセッサID16の実体は、不揮発メモリ（図示しない）に書き込まれた数値・文字データであり、制御ユニット13によって適宜読み出される。

【0065】暗号化復号化ユニット17は、所定のデータを暗号化したり、逆に暗号化されたデータを復号化するための演算ユニットである。例えば、プロセッサID

ロセッサ100に実行させるためには、メモリ管理ユニット15にプログラムIDとそのアドレス情報・属性情報を与える旨のプログラム・コードを制御ユニット13に供給すればよい。メモリ管理ユニット15は、自身のMMUテーブルを参照して、該当するプログラムの実行を許可する仕組みとなっている。下式は、あるプログラムの実行を許可するための疑似コードを記述したものである。

【0060】

【数1】

16のような秘匿性の高いデータは、暗号化復号化ユニット17によって予め暗号化してから、バス・インターフェース11経由でプロセッサ100の外部に出力される。逆に、暗号化されたデータ（例えばソフトウェア供給者から渡されるKey：後述）の復号化も行う。

【0066】暗号化及び復号化の方式は、共通鍵、公開鍵のうちのいずれの方式を採用してもよい。後者の公開鍵方式は、公開鍵と秘密鍵という一組の暗号鍵が用いられ、一方の鍵で暗号化したときには他方の鍵でしか復号化できない。例えば、第3者が公開鍵で暗号化したデータは秘密鍵を持つ本人しか復号化できないので、秘密情報を安全に交換することができる（周知）。

【0067】なお、プロセッサ100を構成するためには、図1に示した以外にも多くのユニット等が必要である。図1から省略されたユニットの例は、整数同士の算術演算と論理演算（AND、OR、NOT）を担当するALU（Arithmetic and Logic Unit）や、実数（浮動小数点等）演算を担当するFPU（Floating-Point Unit）、インストラクションの実行手順を直線的に実装した配線論理回路（PLA：Programmable Logic Array）、入出力用のレジスタなどである。但し、これらは当業者には周知であり、また、本発明の要旨を構成するものではないので、本明細書中では省略している。また、図面の錯綜を回避するため、図中の各ハードウェア・ブロック間の接続も一部しか図示していない点を了承されたい。

【0068】次いで、プロセッサ100によるプログラム保護動作について、図3を参照しながら説明する。

【0069】ソフトウェア供給業者は、保護対象たるソフトウェア・プログラムをフロッピー・ディスク又はCD-ROMなどの記憶媒体に格納した形態で、市場に流通する。このソフトウェア・プログラムには、固有の識

別情報すなわちソフトウェアID（通常はシリアルな番号で構成される）が付されている。ユーザは、フロッピー・ディスク又はCD-ROMなどの記憶媒体をソフトウェア供給業者から購入するという形態で、ソフトウェア・プログラムの限定的な使用権を取得する（記憶媒体の購入は、ソフトウェアの著作権や無制限な使用権を全く意味しない点に充分留意されたい）。そして、ユーザは、記憶媒体を所定のドライブ・ユニット81/82に装填して、ソフトウェア・プログラムをシステム200のハード・ディスク81上にインストールする。

【0070】このインストール作業には、プロセッサID16を共通鍵により暗号化する手続（ステップS100）と、暗号化したプロセッサIDをソフトウェア供給業者に送付する手続（ステップS102）を含んでいる。暗号化処理は、暗号化復号化ユニット17を用いて行われる。送付手続は、ソフトウェア業者に対するKey（後述）の要求を意味する。なお、送付手続は、郵送に依る他、NIC90で接続された外部ネットワーク経由での伝送という形態で行ってもよい。

【0071】ソフトウェア供給業者は、受け取った暗号情報を復号化して、元のプロセッサIDを得る（ステップS104）。そして、このプロセッサIDと、業者自身が手元で管理しているソフトウェアIDとを一体化して、共通鍵により暗号化することによってKeyを生成する（ステップS106）。Keyは、プロセッサIDとソフトウェアIDとを1つのコードとして同時に暗号化したものであり、各々に切り離すことはできない。Keyは、ソフトウェアの使用不能状態を解く鍵としての性質を持ち、郵送あるいはネットワーク経由でユーザに返送される（ステップS108）。

【0072】このKeyを取得した状態で、コンピュータ・システム200上でソフトウェア・プログラムの実行が促されたとする。これは、例えばOS（オペレーティング・システム）のような外部プログラムが”LOA

D MMU…”のような実行許可インストラクションを発行して、メモリ管理ユニット15にプログラムIDとプログラムの格納番地／属性情報がセットすることにより、実現される。

【0073】本実施例の実行許可インストラクションに該当する処理手続は、図3の破線ブロックで囲まれている。実行許可は、メモリ管理ユニット15への各変数のセット以外に所定の認証手続を含んでいる。

【0074】当該インストラクションが発行されると、プロセッサ100は、まず、受け取っているKeyを復号化して、プロセッサIDとソフトウェアIDとを取得する（ステップS110）。復号化処理は、暗号化復号化ユニット17を用いて行われる。

【0075】次いで、プロセッサ100は、復号化して得られたプロセッサIDとソフトウェアIDを、プロセッサ100自身に内蔵されたプロセッサID、及び装填した記憶媒体から得られたソフトウェアIDの各々と比較する（ステップS112）。

【0076】この比較の結果、照合に失敗したならば、メモリ管理ユニット15によるプログラムの実行は拒否される（ステップS114）。他方、照合が成功裡に終わったならば、メモリ管理ユニット15によるプログラムの実行は肯定され（ステップS116）、プログラムが実行される（ステップS118）。

【0077】このような実行許可インストラクションの手続は、例えばマイクロコードROM14中で、上述のようなインストラクションの処理手続を記述することによって実現可能である。

【0078】ステップS110～S116は、”Load MMU …”という単一のインストラクションで実行される。該インストラクションを記述した疑似マイクロプログラム・コードを、以下に示しておく。

【0079】

【数2】

```

mtspr IBAT0, Rs1, Rs2
(Rs1にプログラムのアドレスとフラグを、
Rs2にソフト供給業者から得たキーを、夫々セットする)
ld temp_reg1, Rs2
(キーを取得する)
decrypt temp_reg1
(プロセッサ100に内蔵された共通鍵で復号化する)
cmp temp_reg1, temp_reg3
(IDどうしを比較照合する)
jne unmatched
ld temp_reg1, Rs
(Rsにプログラムのアドレスとフラグをセットする)
st IBAT0, temp_reg1
set flag_good
end_of_instruction
unmatched;
```

```
set flag_bad
end_of_instruction
```

【0080】このようなマイクロプログラムに従った認証手続は、外部プログラムからは全く見えない状態で実行される。本実施例に係る認証手続は、プロセッサ・チップ100内部で完結されるので、プロセッサ100外部からの監視は殆ど不可能なのである。

【0081】第2の実施の形態

本発明の第2の実施例は、図1に示したものと等価なハードウェア構成により具現される。但し、第1の実施例との相違点は、共通鍵ではなく公開鍵方式で暗号化する点にある。図4には、第2の実施例に係るプログラム保護動作をフローチャートにして示している。以下、図4を参照しながら説明する。

【0082】ソフトウェア供給業者は、保護対象たるソフトウェア・プログラムをフロッピー・ディスク又はCD-ROMなどの記憶媒体に格納した形態で、市場に流通する。このソフトウェア・プログラムには、固有の識別情報すなわちソフトウェアID（通常はシリアルな番号で構成される）が付されている。ユーザは、フロッピー・ディスク又はCD-ROMなどの記憶媒体をソフトウェア供給業者から購入するという形態で、ソフトウェア・プログラムの限定的な使用権を取得する（記憶媒体の購入は、ソフトウェアの著作権や無制限な使用権を全く意味しない点に充分留意されたい）。そして、ユーザは、記憶媒体を所定のドライブ・ユニット81/82に装填して、ソフトウェア・プログラムをシステム200のハード・ディスク81上にインストールする。

【0083】このインストール作業には、プロセッサID16を暗号化する手続（ステップS200）と、暗号化したプロセッサID及びプロセッサ100に内蔵された公開鍵をソフトウェア供給業者に送付する手続（ステップS202）を含んでいる。暗号化処理は、暗号化復号化ユニット17が、ソフトウェア供給業者から与えられた公開鍵を用いて行われる。送付手続は、ソフトウェア供給業者に対するKey（後述）の要求を意味する。なお、送付手続は、郵送に依る他、NIC90で接続された外部ネットワーク経由での伝送という形態で行ってもよい。

【0084】ソフトウェア供給業者は、受け取った暗号情報を、自分の秘密鍵を用いて復号化して、元のプロセッサIDを得る（ステップS204）。そして、このプロセッサIDと、業者自身が手元で管理しているソフトウェアIDとを一体化して、再びプロセッサ100の公開鍵を用いて暗号化して、Keyを生成する（ステップS206）。Keyは、プロセッサIDとソフトウェア

IDとを1つのコードとして同時に暗号化したものであり、各々に切り離すことはできない。Keyは、ソフトウェアの使用不能状態を解く鍵としての性質を持ち、郵送あるいはネットワーク経由でユーザに返送される（ステップS208）。

【0085】このKeyを取得した状態で、コンピュータ・システム200上でソフトウェア・プログラムの実行が促されたとする。これは、例えばOS（オペレーティング・システム）のような外部プログラムが“LOAD MMU…”のような実行許可インストラクションを発行して、メモリ管理ユニット15にプログラムIDとプログラムの格納番地/属性情報がセットすることにより、実現される。

【0086】本実施例の実行許可インストラクションに該当する処理手続は、図4の破線ブロックで囲まれている。実行許可は、メモリ管理ユニット15への各変数のセット以外に所定の認証手続を含んでいる。

【0087】当該インストラクションが発行されると、プロセッサ100は、受け取っているKeyを復号化して、プロセッサIDとソフトウェアIDとを取得する（ステップS210）。復号化処理は、暗号化復号化ユニット17が、プロセッサ100自身に内蔵された秘密鍵を用いて行う。

【0088】次いで、プロセッサ100は、復号化して得られたプロセッサIDとソフトウェアIDを、プロセッサ100自身に内蔵されたプロセッサID、及び装填した記憶媒体から得られたソフトウェアIDの各々と比較する（ステップS212）。

【0089】この比較の結果、照合に失敗したならば、メモリ管理ユニット15によるプログラムの実行は拒否される（ステップS214）。他方、照合が成功裡に終わったならば、メモリ管理ユニット15によるプログラムの実行は肯定され（ステップS216）、プログラムが実行される（ステップS218）。

【0090】このような実行許可インストラクションの手続は、例えばマイクロコードROM14中で、上述のようなインストラクションの処理手続を記述することによって実現可能である。

【0091】ステップS210～S216は、“Load MMU…”という単一のインストラクションで実行される。該インストラクションを記述した疑似マイクロプログラム・コードを、以下に示しておく。

【0092】

【数3】

```
mtspr IBAT0, Rs1, Rs2
(Rs1にプログラムのアドレスとフラグを、
Rs2にソフト供給業者から得たキーを、夫々セットする)
ld temp_reg1
```

```

(キーを取得する)
decrypt temp_reg1
(プロセッサ100に内蔵された秘密鍵で復合化する)
cmp temp_reg1, temp_reg3
( IDどうしを比較照合する)
jne unmatched
ld temp_reg1, Rs
(Rsにプログラムのアドレスとフラグをセットする)
st IBAT0, temp_reg1
set flag_good
end_of_instruction
unmatched;
set flag_bad
end_of_instruction

```

【0093】このようなマイクロプログラムに従った認証手続は、外部プログラムからは全く見えない状態で実行される。本実施例に係る認証手続は、プロセッサ・チップ100内部で完結されるので、プロセッサ100外部からの監視は殆ど不可能なのである。

【0094】第3の実施の形態、

本発明の第3の実施例は、プロセッサ100が持つプロセッサIDを用いず、これに代わって時々刻々発生する乱数を用いるとともに、セキュリティ・デバイス95との協働的動作によってプログラム保護を行う点で、上述の第1及び第2の実施例とは相違する。

【0095】図5には、本発明の第3の実施の形態に係るコンピュータ・システム200及びプロセッサ100のハードウェア構成を模式的に示している。以下、各部について説明する。

【0096】プロセッサ100の外部ピンに直結したプロセッサ・バス101は、ブリッジ回路60経由でシステム・バス70に接続されている。ブリッジ回路60は、プロセッサ・バス101とシステム・バス70間の動作速度の相違を吸収するためのデータ・バッファや、メイン・メモリ50へのアクセス動作を制御するためのメモリ・コントローラを含んでいる。

【0097】メイン・メモリ50は、プロセッサ100が実行するプログラム・コードや処理データを一時格納するための読み書き可能メモリであり、通常は、1以上のDRAM(ダイナミックRAM)チップで構成される。メイン・メモリ50上で展開されるメモリ空間は、メモリ管理ユニット15によって管理される。

【0098】システム・バス70には、PCI(Peripheral Component Interconnect)バスやISA(Industry Standard Architecture)バスが該当し、各種周辺装置が相互接続されている。周辺装置には、例えばキーボードやディスプレイなどのコンソール類(図示しない)、ハード・ディスク・ドライブ(HDD)81、フロッピー・ディスク・ドライブ(FDD)82、CD-ROMドライブ83などの外部記憶装置類、外部ネット

ワークと接続するためのネットワーク・インターフェース・カード(NIC)90、シリアル・ポート経由でシリアル・データの入出力を行うためのシリアルI/Oコントローラ91などが含まれる。なお、バス70上の動作は、ロジック・アナライザなどを用いて解析することができる。

【0099】本実施例では、シリアル・ポートにはセキュリティ・デバイス95が外部接続されている。セキュリティ・デバイス95は、保護対象たるソフトウェア・プログラムに付随する性質を持ち、ソフトウェアIDを格納する他、共通鍵又は公開鍵いずれかの方式による暗号化ユニットを含んでいる。セキュリティ・デバイス95の動作については後述する。

【0100】ソフトウェア供給業者は、通常、フロッピー・ディスクやCD-ROMなどの記憶媒体の形態でソフトウェア・プログラムを配布する。各ユーザは、これら記憶媒体を所定のドライブ・ユニットに装填し、HDD81にコピーすることによって、ソフトウェア・プログラムをシステムにインストールすなわち利用可能な状態に置くようになっている。また、最近では、インターネットなどの外部ネットワーク経由でソフトウェア・プログラムが配布されることも多くなっている。

【0101】プロセッサ100は、当業界においてCPU(Central Processing Unit)若しくはMPU(Micro Processing Unit)と呼ばれるものである。プロセッサ100は、通常はマザー・ボード(図示しない)上に搭載される回路チップであり、各入出力装置(図示しない)を制御してデータの授受を行ったり、データを演算処理し、処理結果をメイン・メモリ50上に記憶したりするなどのコンピュータ・システム(図示しない)の中核機能を果たしている。

【0102】図5に示すように、本実施形態に係るプロセッサ100は、バス・インターフェース11と、デコーダ12と、制御ユニット13と、マイクロコードROM14と、メモリ管理ユニット15と、暗号化復号化ユニット17と、乱数発生ユニット18とを含んでいる。

【0103】バス・インターフェース11は、プロセッサ100外部のコンポーネントとの間でインストラクションやデータを授受するためのユニットである。バス・インターフェース11は、プロセッサ・バス17と直結している。なお、プロセッサ・バス17は、ブリッジ回路60を経由してメイン・メモリ50やシステム・バス70と相互接続されている。

【0104】デコーダ12は、バス・インターフェースで受け取ったインストラクションを解釈して、制御ユニット13が理解できる形式に変換するためのユニットである。

【0105】制御ユニット13は、インストラクションを実行する手順を整えるユニットである。実行するインストラクションは、ワイヤード・ロジック（配線論理）で実行するタイプと、マイクロ・プログラムで実行するタイプの2つに大別される。前者の場合は、インストラクションの実行制御に必要な論理を直線的に実装した順序回路が用いられる。また、後者の場合には、インストラクションはマイクロコード（又はピココード）と呼ばれるさらに細かいコードの集合（すなわちマイクロプログラム）で構成されており、該当するマイクロプログラムをマイクロコードROM14から読み出して実行することになる。なお、1つのプロセッサで利用できるインストラクション全体を「インストラクション・セット」と呼ぶ。

【0106】マイクロコードROM14は、インストラクションの実行に必要なマイクロコードを格納した読み出し専用メモリである。言い換えれば、マイクロコードROM14は、各インストラクションの実行手順を定義した不揮発メモリである。例えば、「LOAD」、「STORE」、「MOVE」などのインストラクションは、ある変数をプロセッサ100内のレジスタ（図示しない）に書き込むためのインストラクションであるが、これらの動作の詳細をマイクロコードROM14内で記述することによって定義することができる。

【0107】メモリ管理ユニット（MMU）15は、プログラム・コードやデータをメイン・メモリ上に展開するなど、プロセッサ100のメモリ空間を管理するためのユニットである。メモリ管理ユニット15は、通常、仮想記憶機能とメモリ保護機能という2つの機能を提供している。仮想記憶機能とは、論理アドレスを物理アドレスに変換したり、物理アドレスに該当するブロック（ページ）がメイン・メモリ上に存在しないときに外部記憶装置との間でブロック（ページ）の入れ替え（すなわち「スワッピング」）を行う機能のことである。また、メモリ保護機能とは、1つのプログラムで生じた障害が他のプログラムに影響を及ぼさないようにする機能のことである。

【0108】また、メモリ管理ユニット15は、メモリ空間上に存在する各プログラムのアドレス情報や属性情

報を管理するためのMMUテーブルを備えている。MMUテーブルの構造や機能は図2を参照した上記説明と略同一なので、ここでは説明を省略する。

【0109】あるプログラムをプロセッサ100に実行させるためには、メモリ管理ユニット15にプログラムIDとそのアドレス情報・属性情報を与える旨のプログラム・コードを制御ユニット13に供給すればよい。一般には、「Load MMU…」なるインストラクションによって、実質上、アプリケーションの実行が許可される。

【0110】本実施例では、この「Load MMU…」インストラクションは、マイクロプログラムという形態で実装され、メモリ管理ユニット15への各変数セットの他に、所定の認証手段が含まれている。この認証手段は、マイクロプログラム中で記述されているので、プロセッサ100の外部からは認証手段の存在自体が見えない。認証手段は、プロセッサ・チップ100内部で完結的に実行されるので、この動作を外部から監視することは殆ど不可能である。したがって、「Load MMU…」インストラクションを発行した外部プログラムからは全く見えない状態で、プロセッサ100はこの認証手段を実行することができる。但し、認証手段の詳細は後述する。

【0111】暗号化復号化ユニット17は、所定のデータを暗号化したり、逆に暗号化されたデータを復号化するための演算ユニットである。例えば、プロセッサ100が外部に出力する乱数のように秘匿性のあるデータは、暗号化復号化ユニット17によって予め暗号化してから、バス・インターフェース11経由でプロセッサ100の外部に出力される。逆に、暗号化されたデータ（例えばソフトウェア供給者から渡されるKey：後述）の復号化も行う。

【0112】暗号化及び復号化の方式は、共通鍵、公開鍵のうちいずれの方式を採用してもよい。後者の公開鍵方式は、公開鍵と秘密鍵という一組の暗号鍵が用いられ、一方の鍵で暗号化したときには他方の鍵でしか復号化できない。例えば、第3者が公開鍵で暗号化したデータは秘密鍵を持つ本人しか復号化できないので、秘密情報を安全に送付することができる（周知）。

【0113】乱数発生ユニット18は、時々刻々乱数を発生するためのユニットであり、本実施例ではプロセッサIDの代わりに用いられる。乱数を用いるのは、固定値を用いたのでは、認証手段を複数回重ねて動作を監視することで見破られてしまう虞があるからである。

【0114】なお、プロセッサ100を構成するためには、図5に示した以外にも多くのユニット等が必要である。図5から省略されたユニットの例は、整数同士の算術演算と論理演算（AND、OR、NOT）を担当するALU（Arithmetic and Logic Unit）や、実数（浮動小数点等）演算を担当するFPU（Floating-Point Uni

t)、インストラクションの実行手順を直線的に実装した配線論理回路(PLA: Programmable Logic Array)、入出力用のレジスタなどである。但し、これらは当業者には周知であり、また、本発明の要旨を構成するものではないので、本明細書中では省略している。また、図面の錯綜を回避するため、図中の各ハードウェア・ブロック間の接続も一部しか図示していない点を了承されたい。

【0115】次いで、プロセッサ100によるプログラム保護動作について、図6を参照しながら説明する。

【0116】ソフトウェア供給業者は、保護対象たるソフトウェア・プログラムをフロッピー・ディスク又はCD-ROMなどの記憶媒体に格納した形態で、市場に流通する。このソフトウェア・プログラムには、固有の識別情報すなわちソフトウェアID(通常はシリアルな番号で構成される)が付されている。ユーザは、フロッピー・ディスク又はCD-ROMなどの記憶媒体をソフトウェア供給業者から購入するという形態で、ソフトウェア・プログラムの限定的な使用権を取得する(記憶媒体の購入は、ソフトウェアの著作権や無制限な使用権を全く意味しない点に充分留意されたい)。

【0117】本実施例が、上記第1及び第2の実施例と相違するのは、ソフトウェア・プログラムを格納した記憶媒体と一緒にセキュリティ・デバイス95が配布される点である。ユーザは、ソフトウェア・プログラムをハード・ディスク81上にインストールするとともに、セキュリティ・デバイス95をシリアル・ポートに外部接続する(ステップS300)。この実施例では、Keyの要求は、ソフトウェア供給業者ではなくセキュリティ・デバイス95に対して行われる(後述)。

【0118】コンピュータ・システム200上でソフトウェア・プログラムの実行が促されたとする。これは、例えばOS(オペレーティング・システム)のような外部プログラムが"LOAD MMU..."のような実行許可インストラクションを発行して、メモリ管理ユニット15にプログラムIDとプログラムの格納番地/属性情報がセットすることにより、実現される。

【0119】本実施例の実行許可インストラクションに該当する処理手続は、図6の破線ブロックで囲まれている。実行許可は、メモリ管理ユニット15への各変数の

```
mtspr IBAT0, Rs1, Rs2
(Rs1にプログラムのアドレスとフラグを、
Rs2にソフト供給業者から得たキーを、夫々セットする)
ld temp_reg1, Rs2
(IDを取得する)
gen temp_reg2
(乱数を発生する)
end Cokey, temp_reg2
(共通鍵と乱数をセキュリティ・デバイスに送信する)
receive Encrypteddata, temp_reg2
```

セット以外に所定の認証手続を含んでいる。

【0120】当該インストラクションが発行されると、プロセッサ100は、まず、乱数発生ユニット18が発生した乱数をセキュリティ・デバイス95に送ること、Keyの要求を行う(ステップS302)。乱数の送付はシステム・バス70経由で行われる。但し、乱数は時々刻々変化する性質を持ち、同じ乱数は一回しか使用しないので、バス70の動作をスヌープされてセキュリティが破られる可能性は極めて低い。

【0121】セキュリティ・デバイス95は、受け取った乱数と、自身が格納するソフトウェアIDとを一体化して、共通鍵を用いて暗号化してKeyを生成し、システム・バス70経由でプロセッサ100に送り返す(ステップS304)。

【0122】プロセッサ100は、共通鍵を用いてKeyを復号化して、乱数とソフトウェアIDとに分解する(ステップS306)。復号化処理は、暗号化復号化ユニット17を用いて行われる。

【0123】次いで、プロセッサ100は、復号化して得られた乱数とソフトウェアIDを、プロセッサ100自身が保持する元の乱数、及び装填した記憶媒体から得られたソフトウェアIDの各々と比較する(ステップS308)。

【0124】この比較の結果、照合に失敗したならば、メモリ管理ユニット15によるプログラムの実行は拒否される(ステップS310)。他方、照合が成功裡に終わったならば、メモリ管理ユニット15によるプログラムの実行は肯定され(ステップS312)、プログラムが実行される(ステップS314)。

【0125】このような実行許可インストラクションの手続は、例えばマイクロコードROM14中で、上述のようなインストラクションの処理手続を記述することによって実現可能である。

【0126】ステップS302～S312は、"Load MMU..."という単一のインストラクションで実行される。該インストラクションを記述した疑似マイクロプログラム・コードを、以下に示しておく。

【0127】

【数4】


```

(暗号化されたIDをセキュリティ・デバイスから受信する)
decrypt temp_reg3
( IDを共通鍵で復号化する)
cmp temp_reg1, temp_reg3
( IDどうしを比較照合する)
jne unmatched
cmp temp_reg2, temp_reg3
( 乱数どうしを比較照合する)
jne unmatched
ld temp_reg1, Rs
( Rs にプログラムのアドレスとフラグをセットする)
st IBAT0, temp_reg1
set flag_good
end_of_instruction
unmatched;
set flag_bad
end_of_instruction

```

【0128】このようなマイクロプログラムに従った認証手続は、外部プログラムからは全く見えない状態で実行される。本実施例に係る認証手続は、プロセッサ・チップ100内部で完結されるので、プロセッサ100外部からの監視は殆ど不可能なのである。

【0129】第4の実施の形態

本発明の第4の実施例は、図5に示したものと等価なハードウェア構成により具現される。但し、第3の実施例との相違点は、共通鍵ではなく公開鍵方式で暗号化する点にある。図7には、第4の実施例に係るプログラム保護動作をフローチャートにして示している。以下、図7を参照しながら説明する。

【0130】ソフトウェア供給業者は、保護対象たるソフトウェア・プログラムをフロッピー・ディスク又はCD-ROMなどの記憶媒体に格納した形態で、市場に流通する。このソフトウェア・プログラムには、固有の識別情報すなわちソフトウェアID（通常はシリアルな番号で構成される）が付されている。ユーザは、フロッピー・ディスク又はCD-ROMなどの記憶媒体をソフトウェア供給業者から購入するという形態で、ソフトウェア・プログラムの限定的な使用权を取得する（記憶媒体の購入は、ソフトウェアの著作権や無制限な使用权を全く意味しない点に充分留意されたい）。

【0131】本実施例では、ソフトウェア・プログラムを格納した記憶媒体と一緒にセキュリティ・デバイス95に配布される。ユーザは、ソフトウェア・プログラムをハード・ディスク81上にインストールするとともに、セキュリティ・デバイス95をシリアル・ポートに外部接続する（ステップS400）。Keyの要求は、ソフトウェア供給業者ではなくセキュリティ・デバイス95に対して行われる（後述）。

【0132】コンピュータ・システム200上でソフトウェア・プログラムの実行が促されたとする。これは、

例えばOS（オペレーティング・システム）のような外部プログラムが”LOAD MMU…”のような実行許可インストラクションを発行して、メモリ管理ユニット15にプログラムIDとプログラムの格納番地／属性情報がセットすることにより、実現される。

【0133】本実施例の実行許可インストラクションに該当する処理手続は、図7の破線ブロックで囲まれている。実行許可は、メモリ管理ユニット15への各変数のセット以外に所定の認証手続を含んでいる。

【0134】当該インストラクションが発行されると、プロセッサ100は、まず、乱数発生ユニット18が発生した乱数とプロセッサ100の公開鍵をセキュリティ・デバイス95に送ることで、Keyの要求を行う（ステップS402）。乱数の送付はシステム・バス70経由で行われる。但し、乱数は時々刻々変化する性質を持ち、同じ乱数は一回しか使用しないので、バス70の動作をスヌープされてセキュリティが破られる可能性は極めて低い。

【0135】セキュリティ・デバイス95は、受け取った乱数と、自身が格納するソフトウェアIDとを一体化して、公開鍵を用いて暗号化してKeyを生成し、システム・バス70経由でプロセッサ100に送り返す（ステップS404）。

【0136】プロセッサ100は、自身の秘密鍵を用いてKeyを復号化して、乱数とソフトウェアIDとに分解する（ステップS406）。復号化処理は、暗号化復号化ユニット17を用いて行われる。

【0137】次いで、プロセッサ100は、復号化して得られた乱数とソフトウェアIDを、プロセッサ100自身が保持する元の乱数、及び装填した記憶媒体から得られたソフトウェアIDの各々と比較する（ステップS408）。

【0138】この比較の結果、照合に失敗したならば、

メモリ管理ユニット15によるプログラムの実行は拒否される(ステップS410)。他方、照合が成功裡に終わったならば、メモリ管理ユニット15によるプログラムの実行は肯定され(ステップS412)、プログラムが実行される(ステップS414)。

【0139】このような実行許可インストラクションの手続は、例えばマイクロコードROM14中で、このようにインストラクションの処理手続を記述することによ

```

mtspr IBAT0, Rs1, Rs2
  (Rs1にプログラムのアドレスとフラグを、
  Rs2にソフト供給業者から得たキーを、夫々セットする)
ld temp_reg1, Rs2
  (IDを取得する)
gen temp_reg2
  (乱数を発生する)
end Pubkey, temp_reg2
  (公開鍵と乱数をセキュリティ・デバイスに送信する)
receive Encrypteddata, temp_reg2
  (暗号化されたIDをセキュリティ・デバイスから受信する)
decrypt temp_reg3
  (IDを秘密鍵で復合化する)
cmp temp_reg1, temp_reg3
  (IDどうしを比較照合する)
jne unmatched
cmp temp_reg2, temp_reg3
  (乱数どうしを比較照合する)
jne unmatched
ld temp_reg1, Rs
  (Rsにプログラムのアドレスとフラグをセットする)
st IBAT0, temp_reg1
set flag_good
end_of_instruction
unmatched;
  set flag_bad
  end_of_instruction

```

【0142】このようなマイクロプログラムに従った認証手続は、外部プログラムからは全く見えない状態で実行される。本実施例に係る認証手続は、プロセッサ・チップ100内部で完結されるので、プロセッサ100外部からの監視は殆ど不可能なのである。

【0143】追補

以上、特定の実施例を参照しながら、本発明について詳解してきた。しかしながら、本発明の要旨を逸脱しない範囲で当業者が該実施例の修正や代用を成し得ることは自明である。すなわち、例示という形態で本発明を開示してきたのであり、限定的に解釈されるべきではない。本発明の要旨を判断するためには、冒頭に記載した特許請求の範囲の欄を参酌すべきである。

【0144】

【発明の効果】以上詳記したように、本発明によれば、

って実現可能である。

【0140】ステップS402～S412は、“Load MMU …”という単一のインストラクションで実行される。該インストラクションを記述した疑似マイクロプログラム・コードを、以下に示しておく。

【0141】

【数5】

コンピュータ・システムに提供されるソフトウェア・プログラムの不正な使用・実行を防止する優れた技術を提供することができる。

【0145】また、本発明によれば、プログラムの不正実行防止機能を備えた優れたプロセッサ、プログラムの不正実行を防止するためにプロセッサが実行するインストラクション、及びプログラムの不正実行防止方法を提供することができる。

【0146】また、本発明によれば、外部から監視・変更することができないプログラムの不正実行防止機能付き優れたプロセッサ、プロセッサが実行するインストラクション、及びプログラムの不正実行防止方法を提供することにある。

【0147】また、本発明によれば、ソフトウェア供給者がソフトウェア・プログラムの使用権をただ1つのプ

ロセッサにのみ許可することができる、優れたプログラムの不正実行防止方法を提供することができる。

【図面の簡単な説明】

【図 1】図 1 は、本発明の第 1 の実施の形態に係るコンピュータ・システム及びプロセッサ 100 のハードウェア構成を模式的に示した図である。

【図 2】図 2 は、MMU テーブルの構造や機能を模式的に示した図である。

【図 3】図 3 は、本発明の第 1 の実施の形態に係るプログラム保護動作を示したフローチャートである。

【図 4】図 4 は、本発明の第 2 の実施の形態に係るプログラム保護動作を示したフローチャートである。

【図 5】図 5 は、本発明の第 3 の実施の形態に係るコンピュータ・システム及びプロセッサ 100 のハードウェア構成を模式的に示した図である。

【図 6】図 6 は、本発明の第 4 の実施の形態に係るプロ

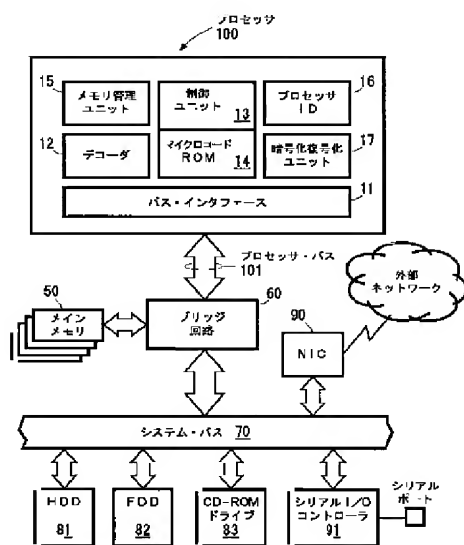
グラム保護動作を示したフローチャートである。

【図 7】図 7 は、本発明の第 5 の実施の形態に係るプログラム保護動作を示したフローチャートである。

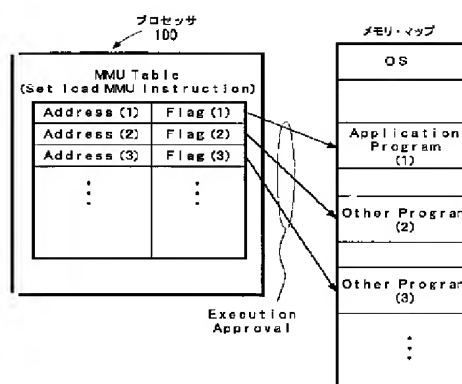
【符号の説明】

11…バス・インターフェース、12…デコーダ、13…制御ユニット、14…マイクロコード ROM、15…メモリ管理ユニット (MMU)、16…プロセッサ ID、17…暗号化復号化ユニット、50…メイン・メモリ、60…ブリッジ回路、70…システム・バス、81…ハード・ディスク・ドライブ (HDD)、82…フロッピー・ディスク・ドライブ (FDD)、83…CD-ROM ドライブ、90…ネットワーク・インターフェース・カード (NIC)、91…シリアル I/O コントローラ、95…セキュリティ・デバイス、100…プロセッサ、200…コンピュータ・システム。

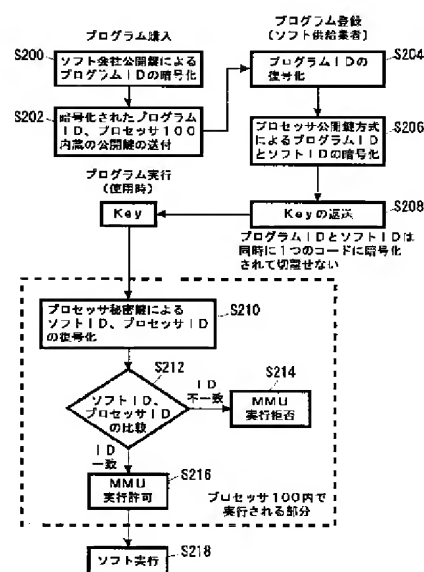
【図 1】



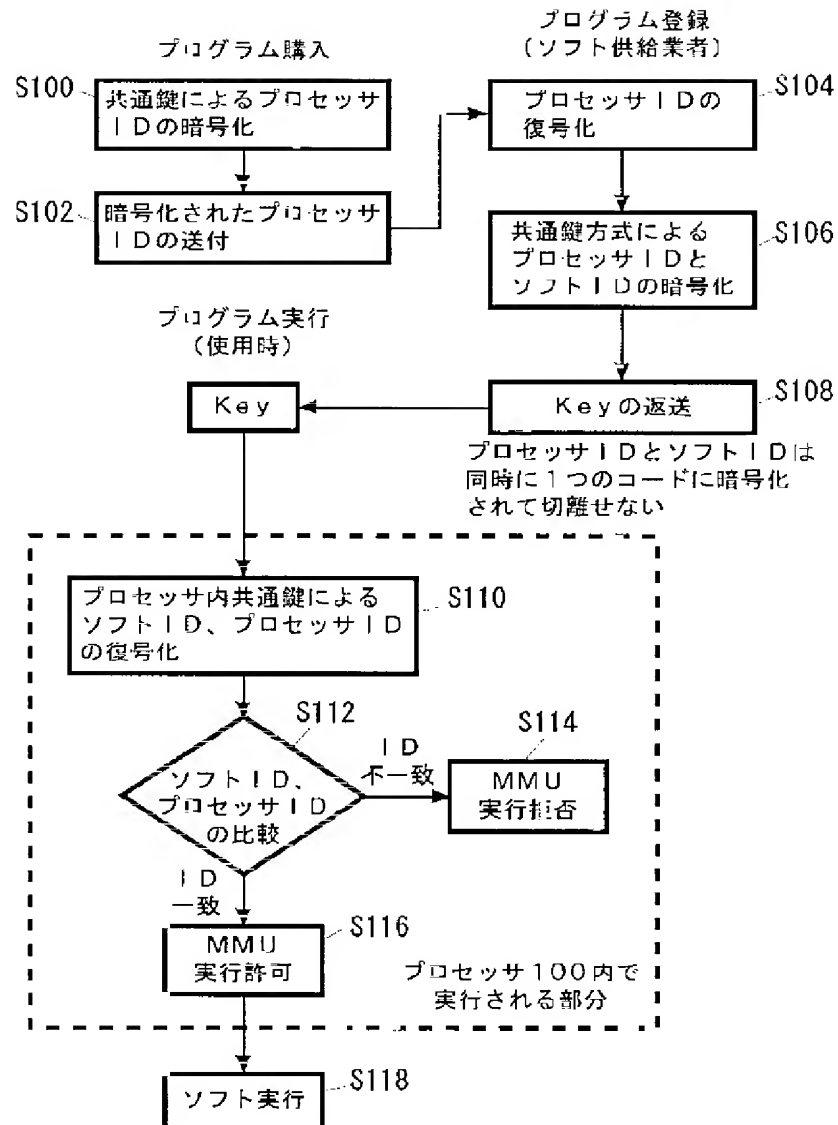
【図 2】



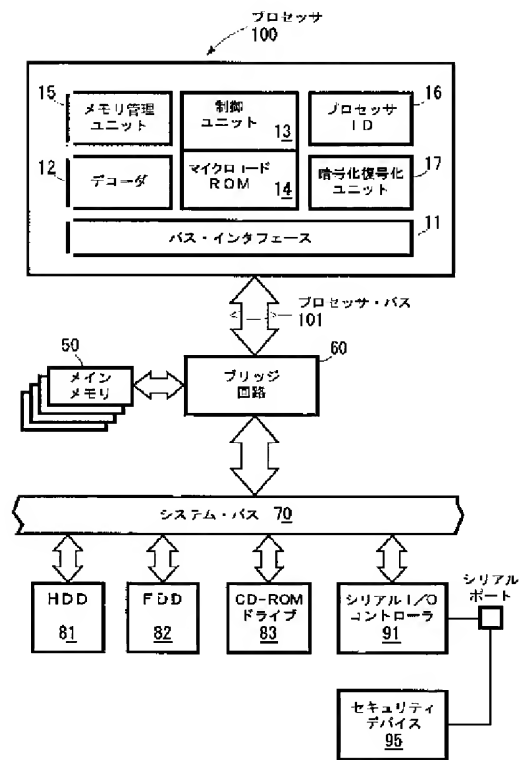
【図 4】



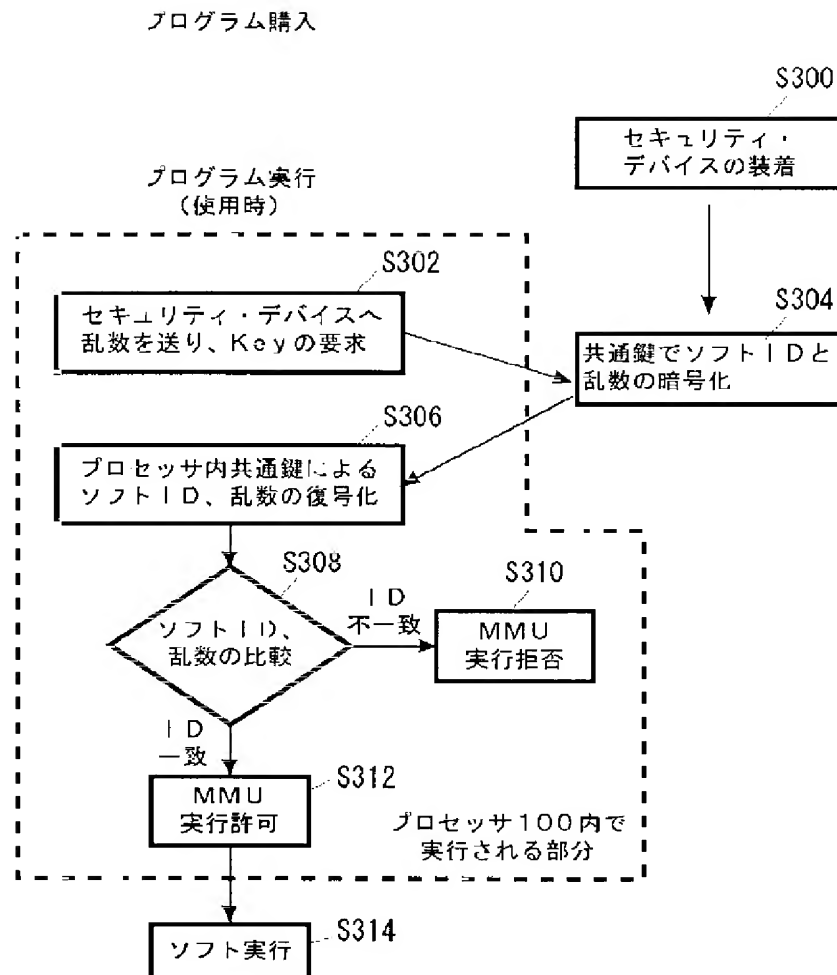
【図3】



【図 5】



【図6】



【図7】

